Remote teaching and learning of software testing using active methodologies in the COVID-19 pandemic context

Isaac Souza Elgrably
Graduate Program in Computer Science (PPGCC), Institute of Exact and Natural Sciences (ICEN)
Federal University of Pará (UFPA)
Belém/PA, Brazil
isaacelgrably@gmail.com

Sandro Ronaldo Bezerra Oliveira
Graduate Program in Computer Science (PPGCC), Institute of Exact and Natural Sciences (ICEN)
Federal University of Pará (UFPA)
Belém/PA, Brazil
srbo@ufpa.br

***Abstract*— This Research for the Practice of Full Paper shows that in the face of the COVID-19 pandemic, the immense challenge of continuing the teaching-learning process without face-to-face classes arose. This required a rapid shift to distance education in order to enable continuity of university activities. Distance education still needs the help of other factors, as some subjects, for example software testing, may need other forms of active approaches and methodologies to make their teaching and learning process more efficient. In teaching some computer science subjects, there is a need for learning through the application of practical activities. The use of distance learning entails an even greater aggravation of this need, given the numerous limitations that exist in this modality, requiring good planning and rigorous monitoring before, during and after classes. Thus, this paper aims to identify and discuss the findings, good practices, recommendations and perceptions used in a software testing subject, in addition to analyzing the motivations and perceptions of students in a distance learning environment.**

Keywords—software testing, activities methods, remote teaching-learning, covid-19 pandemic.

I. INTRODUCTION

Due to the COVID-19 pandemic, most universities in the world had to adapt to the scenario of social detachment and isolation. In order to avoid a complete halt in university activities, some institutions have joined remote education. Thus, the construction of reports on preliminary remote teaching experiences is crucial to disseminate knowledge about the difficulties faced, the lessons learned and the students' perception of the strategies adopted [1].

This paper describes the authors' experience in teaching a subject in a graduate program at a Brazilian university during the COVID-19 pandemic. Based on the situations that have occurred, we propose a set of practices that can help other professors and researchers to build subjects in the future. This Software Quality subject focused on teaching the use of a curriculum aimed at teaching software tests [2] and using a teaching plan aimed at using active teaching methodologies [3] adapted to the reality of remote teaching.

The work also presents the students' perception of the teaching methodology and approaches adopted. The students also reported their perceptions about how different methodologies enrich knowledge and can be carried out in the remote context.

A Software Quality graduate course was the context of the work presented and had the following Research Question (RQ): Is it possible to guarantee motivation in a software testing subject that uses active teaching methodologies, using remote teaching?

To answer this RQ, a classroom was developed with synchronous and asynchronous learning and supported by several technological resources and different methodology active teaching that were described in the following sections. At the end of the course, the results of a questionnaire on the perception of motivation in the subject, the teaching approaches and the influence of remote teaching were collected. These results served as guidelines for the authors of the work to adapt future solutions and anticipate future problems that may happen during a subject carried out remotely.

This paper does not aim to make a comparison of classroom teaching with remote teaching, nor analyze the learning of the subject presented. The scope of the work is to present a set of experiences found in the experiment and present results of motivation and acceptance of active teaching methodologies, some of them being group activities in a difficult remote learning scenario in a pandemic context.

In addition to this introductory section, this paper follows the following structure: in Section II the theoretical basis for the construction of this work is presented, contextualizing the main topics present in it, in Section III the research methodology is described, indicating how it was conceived to participation of students and how their results were analyzed, Section IV presents the subject of software testing that was taught, how it was built, adaptations that were necessary and how its execution worked, in Section V a discussion about the results that have been achieved, and in Section VI the conclusion of this work is presented, indicating the contributions of this work, its limitations and the future works that will continue this work.

II. BACKGROUND

This section aims to provide a theoretical basis for the concepts used in this work.

A. Education in Remote Teaching

As seen in [4], engagement and motivation become a major challenge in distance education in computer science and software engineering courses.

Remote education has undergone some adaptations due to current needs, some research on effective methods in the remote education modality is relatively new in the context of emergency crises [1].

It is possible to separate the form of content transmission in remote education into two categories: synchronous

learning and asynchronous learning. Synchronous learning involves participation by everyone involved while the teacher teaches the class. The environment for this modality to occur is usually interactive, using video calls, chat sessions or using synchronous learning tools [5]. On the other hand, asynchronous learning tends to be more flexible, since it does not require the simultaneous presence of everyone involved [5].

Some recent works in the literature carried out in Brazilian universities present us with some challenges in the context of remote learning in a pandemic situation, such as Trinta, Rego and Viana [6], which point out that the learning model using remote learning is far from ideal, due to several factors such as the lack of better planning, improvisation of the solutions used and the participants' emotions in the face of the pandemic. On the other hand, that of Lelli et al. [7] that even with stimuli on serious game, gamification and other active teaching methodologies, many students face several physical and psychological difficulties to follow Computer Science courses at that time.

They served as inspirations, the work [1] talks about an account of an experience of teaching usability and UX subjects using a methodology based on synchronous meetings and asynchronous activities using practical tasks and the work [8] that the authors had developed a Software Engineering subject to be taught a face-to-face learning experience and had to adapt it for remote teaching.

B. Teaching of Software Testing

With a greater need in the job market to have highly qualified software testers, many university software engineering (SE) and computer science (CS) programs around the world have started to include software testing in their education curricula [9]. Tests are one of the most valued programming practices, in any type of software. It is also one of the practices that has undergone the most changes in recent years [9].

In the systematic mapping carried out by Cunha et al, [10], among the sub-areas of Software Engineering, the one of software tests was the second sub-area with more works found. However, there is still a considerable deficiency in the teaching of software tests, where there are still complaints related to the teaching of tests in education, as its focus is on theory and lack of practice to show students how the concepts should be applied [11].

One of the results presented by Garousi et al. [12] states that there is a misalignment between what researchers and the job market have as a priority over software testing. While researchers tend to be more interested in theoretically challenging issues, for example, research-based test case design, professional test experts in practice look for options to improve the effectiveness and efficiency of tests.

The perception about the teaching of Software Testing has been refined over the years and the needs of the job market pressure universities to adapt to better train their students, thus composing a curriculum and a teaching plan to motivate students and adding knowledge is a challenge to be overcome by universities, especially in a context of distance due to the pandemic.

C. Active Teaching Methodologies

To consolidate a good learning environment it is necessary that for each different one the objective to be achieved must apply specific methodologies. For Morán [13], active methodologies are starting points to move towards more advanced processes of reflection, cognitive integration, generalization and re-elaboration of new practices.

Active methodologies, when taken as a basis for planning learning situations, can contribute significantly to the development of student autonomy and motivation as it favors the feeling of belonging and co-participation in the learning process [14].

The SBC (Brazilian Computer Society) 2017 Curricular Guide [15] indicates that in computer science courses whenever possible, active methodologies should be employed, so that the student spends more time in activities in which he / she is a protagonist in the teaching and learning process.

The Brazilian Ministry of Education (MEC) [16] proposes that the teaching methodology for computer courses should be centered on the student as a subject of learning and supported by the teacher as a facilitator of the teaching-learning process and that the teacher should also show the applications of theoretical content, being a mediator, stimulating competition, communication, provoking teamwork, motivating students to study, guiding reasoning and developing communication and negotiation skills.

So it is seen that the regulations and guides to support the teaching of computer courses at Brazilian universities encourage the use of active methodologies to support the transfer of knowledge and the understanding of the teachings for students.

III. RESEARCH METHODOLOGY

As mentioned earlier, this work arose through the need to teach a subject focused on teaching software testing using a model curriculum of tests [2] and a teaching plan aimed at practical teaching of software tests using active teaching methodologies [3], such as: Practical project [17], Dojo [3], Playful Teaching [3, 18] and Discussions of practical cases [19], thus forming a teaching methodology for a software testing subject. Figure 1 shows the methodology applied in this work.

After the construction of the teaching plan, some adaptations were necessary to apply it remotely while maintaining the guidelines of the teaching plan [3]. After the adaptation began, its execution in a subject of Software Quality in a Graduate Program in Computer Science of a Brazilian Federal University.

The way of transferring the contents of the curriculum of testing in the subject will be explained in Section IV. At the end of each class, the authors of the work met and evaluated the events that occurred in that class and took notes on a virtual spreadsheet what could be done to improve that class or activity. These findings will be presented in Section V.

Other results that came from the students' perception were collected at the end of the course in a questionnaire with quantitative and qualitative questions and sent blindly, in which the students were informed that it would not be known who would have authored the answer to the questionnaire. Finally, the data collected in the

questionnaires were analyzed by the authors and will be explained in Section VI.

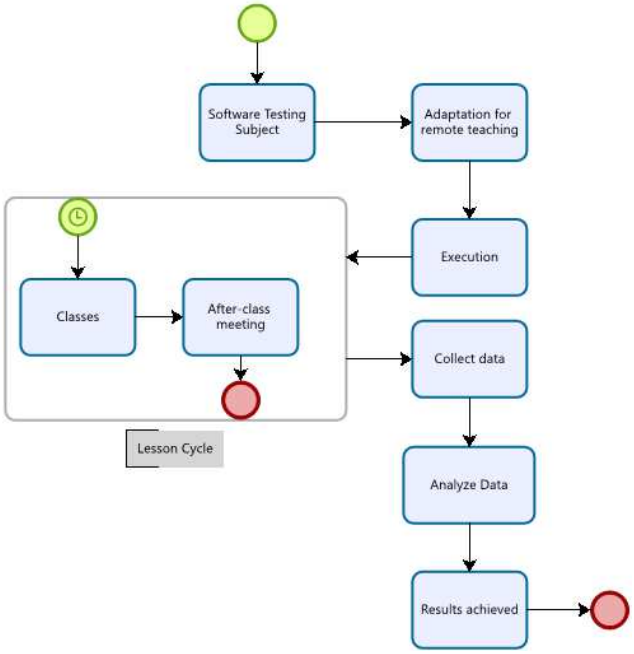

Fig. 1. Methodology applied in the research.

IV. THE SOFTWARE TESTING SUBJECT

The Software Quality subject adapted to the teaching of software testing [2, 3], which was used in the experiment described in this work, is offered semiannually to regular and special graduate students (masters and doctorate courses) and undergraduate students in Computer Science. It is a subject considered optional for the training of these students in Software Engineering. Currently, the course is taught by a professor with extensive academic and professional experience.

The students of this professor actively act in the administration of this subject based on monitoring, in addition to teaching classes and promoting activities with the class. This course has a 60 class hour with 2 weekly meetings.

A. Curriculum and Teaching Plan for Software Testing adopted

According to the ACM / IEEE (Association for Computing Machinery / Institute of Electrical and Electronics Engineers) guide [20], there is no single formula for success in developing an Information Technology curriculum. The curriculum [2] used in this experiment takes advantage of the construction of knowledge by competences. According to Sacristán et al. [21], a competence is the ability of a student to use means, such as: knowledge, skills, attitudes and values, to perform in complex real situations.

The subject was carried out at a Brazilian Federal University, so it was decided to use the existing competences in MEC [16], as it is the regulator of the curricular guidelines for the bachelor's and undergraduate courses in computing together with those of the SBC Guide [20]. It was divided into four teaching units: Software Engineering, Software Construction, Software Quality and Testing. Each of these units had its content [2].

The teaching plan [3] used is an adapted version of the work in which different teaching approaches have their own learning objectives, based on Bloom's Revised Taxonomy [22], which is composed of knowledge dimension (Fatual, Conceptual, Procedural and Metacognitive) and the dimension of the cognitive process (Remember, Understand, Apply, Analyze, Evaluate and Create) to build the knowledge of the existing contents in each teaching unit.

The teaching approaches that were used in this subject were: expository lecture, list of exercises, dojo randori, dojo kake, discussion of practical cases and practical project. This can best be seen in Table I.

TABLE I. TEACHING APPROACHES, METHODOLOGICAL PROCEDURES AND

Teaching Approach	Methodological Active Procedures	Active Methodologies
Dialogued Expositive Class	There is an exhibition of content with the active participation of students. The teacher has the role of mediator for students to question, interpret and discuss the object of study. The support material is made available in advance so that the student can study the content in advance.	Flipped Classroom
List of Exercises	The list of exercises available in a class is corrected as a group at the beginning of the subsequent class, enabling an exchange of knowledge for learning.	Flipped Classroom
Dojo Randori	It is a practice of teaching programming in a safe environment, eading to peer learning. The student will learn by trial and error, with the help of his / her pairs. All students participate in the activity, as a pilot, co-pilot or audience [3].	Flipped Classroom; Learning in Pairs
Dojo Kake	In this practice, multiple pairs work in parallel to carry out the activity. Each pair will carry out a list of challenges and the teachers helped, answering questions about the activity.	Flipped Classroom; Learning in Pairs
Discussion of Practical Cases	There is the presentation of reports of existing experiences in the software development industry or in project labs at universities in order to present students with problems that usually happen in real environments [18].	Discussion of Practical Cases; Learning in Pairs
Practical Project	This practice aims the learning focused on delivering experience that simulates the real world for the students, making them come to acquire skills that are important in the challenges that exist in the job market.	Practical Project; Learning in Pairs

In addition to the methodological procedures mentioned in Table I, each of the learning approaches needs to deliver its own learning objectives, the same teaching approach performed differently may have different learning objectives following Bloom's Revised Taxonomy [22].

B. Necessary Adaptations

The adaptation of an experiment to teach a software testing subject designed to use active teaching methodologies had a certain degree of difficulty.

First, a virtual classroom was created in *google classroom* [1] to serve for content management, storage of materials and communication between students and teachers, the MEC requested that all synchronous classes should be recorded and made available to students, if they could not participate in a class. A group was also created with students on the whatsapp tool for general questions, notices and materials that can also be inserted. So, the experiment had not been idealized to be carried out with Remote Teaching, it was necessary to make several adaptations in each teaching approach to be able to be carried out.

For the teaching approach of dialogued expository class, the following actions were taken: (i) Use of a remote videoconferencing tool (*google meet*[2]), (ii) Decision of all classes in this aspect to be given synchronously, in order to maintain the discussion and the students to be able to participate actively in the class, (iii) When starting the class, only the teacher who was teaching the class kept the audio and video open, and (iv) Another teacher managed the participation via chat using the tool, answering questions via chat and at other times opening the possibility for a student to participate and expose comments, doubts and feed the dialogue. There is also the possibility for other students to also participate in the dialogue in an organized manner.

For the teaching approach using the list of exercise, the following actions were taken: (i) The lists were made available via the *google form* [3] linked to the google classroom, (ii) Students had until the next class to answer the list and at the beginning of the next class doubts and discussions were made, and (iii) Students could use the classroom activity feature to initiate a chat with teachers to answer questions about activities.

For the teaching approach of dojo randori, the ideas seen in [3] of using the Scratch tool and the participation of all students in the activity were maintained, the following actions were taken: (i) The dojo was carried out using the Scratch online editor platform maintained by MIT - Massachusetts Institute of Technology, (ii) Before the Dojo class, a tutorial video on using the online editor at *Scratch*[4] was added to the subject's classroom and demonstrated features and how it could be done to share the project with another user, (iii) A manual on the use of Scratch functionalities and on the TDD - Test Driven Development cycle [23] was added to the subject's classroom, and (iv) Students were asked to register on the MIT platform before the day of the activity.

For the teaching approach of dojo kake, the following actions were taken: (i) The development environment and settings via google classroom were made available to students, (ii) A tutorial on using the development environment and a screen recording tool was created and made available, (iii) The number of students for group was increased from two to three, (iv) A new operation was defined for the practice, each of the teams would create a video conference room at google meet and the teachers divided the rooms between them, and (v) Each of the teams should, at the end of the activity, send the developed source code and the video of their team for the teachers to analyze both the development of the practice and the source code.

For the teaching approach to the discussion of practical case and the presentation of practical project, the same measures were taken as the lectured expository class. These were the main adaptations necessary in a moment prior to the beginning of the subject.

C. Execution

The experiment was carried out in the second semester of 2020 in a graduate course in Computer Science at a Brazilian Federal University. The teaching plan presented by the graduate program already contained the curriculum and the approaches that would be used. In the first class, the learning nature and objectives for students were clarified.

The subject had 17 students enrolled, but 2 of them dropped out of the subject with a total of 15 students. The subject was taught in 32 classes with the support of four professors and fulfilling the entire content of the curriculum [2]. Table II shows the types of the class.

TABLE II. APPROACHES, CLASS TYPES AND QUANTITY OF ACTIVITIES

Teaching Approaches	Class Types	Quantity of activities
Dialogued Expositive Class	Synchronous	21
List of Exercises	Asynchronous	3
Dojo Randori	Synchronous and Asynchronous	1 (two days)
Dojo Kake	Synchronous and Asynchronous	2
Discussion of Practical Cases	Synchronous	2
Practical Project	Synchronous and Asynchronous	2

The amount of activities is not the same as the number of classes, as some of them were carried out asynchronously and discussed in part of the class, for example: on a day of class, a list of exercises could be discussed and later given a dialogued expositive class.

All twenty-one dialogued exposite classes were given in synchronous forms, with a presence of around 92% by the students. The lists of exercises passed at the end of the class were all done individually and in two moments a student delayed delivery and had part of the score deducted. These questions had important aspects present in the dialogued classes.

The dojo randori should have happened in one class, but during the activity the professors decided to extend it for another class increasing the complexity of the challenge. The asynchronous part were the tutorials and manuals that the students had access via google classroom. This was an activity in a group with the whole class with the same goal.

There were two dojo kake practices, a first to maintain a system developed by one of the professors and the second one was an exchange of source codes between the teams with

[1] https://classroom.google.com/u/0/h

[2] https://meet.google.com/

[3] https://docs.google.com/forms/u/0/

[4] https://scratch.mit.edu/

the aim of adding unit tests, expanding test coverage, design improvements and problem fixes when needed. Like the dojo randori, the asynchronous part was made from tutorials and the activity was carried out in group of three students.

The discussions of practical cases took place in two synchronous activities, in which a guest professor presented techniques of Exploratory Tests from Charts and passed a challenge for students to find defects in a web page and discuss the findings in class. The other class was one of the professors presenting the *Jmeter* [5] tool in conjunction with the *BlazeMeter*[6] plugin presenting a case in the classroom of a load and performance testing in an automated way. The students participated with ideas for building test cases and configuration parameters.

The practical project activities were carried out in two moments of presentation and three mentoring classes, in which the professors were at the disposal of the teams to answer questions and advise on the activity and in the spare time the teams carried out the activities. The practical project was an evolutionary activity, and in the first moment the students, analyzing the interface of an implemented software, performed an ad-hoc test analysis to collect errors, failures and new requirements for the system and later presented it to the professors. In a second step, students through what they collected in the analysis of ad-hoc testing, built a prototype and a test plan and presented the results to the professors.

It is important to comment that this work that is being presented, does not aim to analyze the results of the grades obtained by the students, but rather to comment on how the subject took place in a remote environment and how the students will evaluate the activities, tools and motivation about the subject.

V. Findings from the Experience Application

With the introduction of remote education almost imminently with a very short adaptation period, adapting a subject using active teaching methodologies has become a very challenging scenario. The curriculum content of the course was designed to be taught in a synchronous and face-to-face manner. According to Kanji and Grundy [6], remote learning is usually more suitable for asynchronous learning approaches. So, even with the adaptations that were made previously, the professors of this work at the end of each day of class met to discuss and conclude if there was any new challenge and how to get around it. Then one of the professors filled out an electronic spreadsheet storing the challenges and how to solve them and added good practices found.

A. Challenges Found

Some challenges arose during the execution of the course and were being adapted for later classes. In this subsection we will indicate them and present the perception that the professors had about each one.

1) Dynamics with the whole class: During the Dojo Randori activity, the flow of the activity ended up being hampered by the need for changing roles (pilot, co-pilot and audience) among the students. This ended up necessitating the addition of one more class to complete the activity.

2) Need for advance warnings: For the other Dojo approaches it was decided that warnings or explanations of the activities to be performed in the next class should always be communicated in this class. In this way, all students prepare for activity, being able to clarify doubts or request support from one of the professors in order to avoid any waste of time during the execution of activities. Thus, in the practical application class, all the factors necessary for the application to occur are reinforced.

3) Difficulty for collaboration: It was noticed that collaborative actions positively influence the performance in carrying out group activities involving all students interacting via audio than via chat. However, in remote education we have great obstacles to organize discussions and activities with the whole class like Dojo Randori, there was a strong concentration of practice in only a few students.

4) Preparation of environments in different operating systems: As there was no availability of a laboratory for activities to be carried out in person, providing only the tools and an installation tutorial was not enough. Different operating systems and configurations specific to the students ended up increasing the difficulty for this task. One of the professors was available to answer questions and, when necessary, guide the installation and configuration. For future subjects, the authors intend to build the tool and activities environment in a docker container and make it available to students.

5) Control of activity time: It was observed that the activities carried out during the class require flexible times, given that in some tasks, the addition of time, in addition to the programmed time, becomes positive, enabling greater interaction and development between students. Similarly, the decrease in time must also be observed, as some tasks can be performed quickly, causing idleness in the class. In both cases, it is necessary for professors to interfere in observing these needs.

B. Good Practices Used

During the course, conversations with professors, participation in conferences and monitoring of the scientific community, some good practices and adaptations to the needs of remote teaching were captured by the professors and added to the experiment. In this subsection we will indicate them and present the perception that professors had about each one.

1) Communication: A well-defined communication about the applied procedures allowed the students to solve the group activities more quickly even though there was the circumstance of knowing the virtues of each student. This effective communication made it possible to more quickly identify the profiles of each student and thus evolve their processes, reaching their goals.

2) Synchronous class: It was observed that this way there is more interaction between professor and students, because as the subject is approached, students are able to clarify their doubts, question about certain concepts, examples are more directed to the question, the doubt what is being clarified is observed by all students and even help from other students may occur.

[5] https://jmeter.apache.org/
[6] https://www.blazemeter.com/

3) Instant feedback: It is directly related to the synchronous class modality, as this allows students to question, suggest and talk via chat, or even losing permission to talk via audio when certain issues or context are more difficult to describe quickly in a textual way, enabling the interaction of several students on the same issue.

4) Auxiliary professors: With the participation of an auxiliary professor, it is more appropriate to evaluate the performance of the students, as any situation, whether positive or negative, is noted for possible improvements. In addition, the auxiliary professor is able to interact in moments of discussion via chat or audio when necessary to reinforce certain concepts, explain in more detail experienced situations appropriate to the theme and provide guidance on activities.

5) Analysis meeting of the class: At the end of each class, the professors met to discuss many factors that were perceived during the class in order to improve it for the next class, be it didactics, material, examples covered, etc.

6) Make an example about the procedures: Demonstrate the procedures at the beginning of the activity to facilitate understanding about the use of tools used and the procedures as a whole. It is important that students have prior contact with the tool and read about the materials involved in the activities. This contact can be made via manuals, tutorials or asynchronous classes.

7) Respect: Respect for all defined ideas, opinions, suggestions, recommendations, strategies allowed better performance in the activities to happen, quickly achieving the expected result of the group work. In addition, it was possible to see that respect facilitated changes or adjustments in established processes and strategies.

8) Reflection: The brief moments of reflection allowed us to align the suggestions to be followed with the procedures necessary to achieve the common goal of the team. From moments of reflection, students were able to distinguish among a set of ideas which were well applied and appropriate to the procedure or process used. Students in the role of co-pilot and audience at the Dojos had these moments of reflection in the activity.

9) Facilitator: The interference of a professor is essential when the process deviates from the objectives set for the activity in order to ensure that the students remain aligned and activities are adhering to that objective.

10) Classes dialogued with contemporary *examples:* It was observed that when classes involve examples of contemporary life after the presentation of concepts, techniques, approaches in general, students interact more by asking questions and giving suggestions on how these factors can be applied. Because there are always students with different profiles and previous knowledge, it is extremely important to give attractive and contemporary examples so that there is a greater participation of students.

11) Check-list of processes: In the virtual classroom process, different resources and processes are required. Thus, it is necessary to list the procedures and they must be strictly followed and controlled, in order to avoid undesirable results or rework.

12) Guided example: It was noticed that by presenting an example and, together with the class, working out the solution, the students' participation increased. After the example, some students presented results of their attempts at solution using other methods or tools. This good practice worked very well in conjunction with the discussion of practical case approach, in which the class immersed themselves in the context of the class and helped guide it.

VI. RESULTS

The RQ of this paper is about whether, through the different perceptions of each student, it was possible to guarantee a motivation in a software testing subject that uses active teaching methodologies, using remote teaching. In this context, the following were evaluated: relevance, adequacy, teaching method, motivation and remote teaching. A total of 10 questions, all on a Likert 5 scale. For the items, the internal consistency coefficient of Cronbach's Alpha [24] was considered "Acceptable" with $\alpha = 0.716$, allowing the results to be related for an analysis. Table III shows the questions asked in the questionnaire and for each question it was mandatory for students to make comments for a qualitative analysis of the choices.

TABLE III. QUESTIONS INCLUDED IN THE QUESTIONNAIRE

Item	Question
1	In your view, was the content taught in the Software Testing subject relevant?
2	In your view, were the approaches to teaching the contents of Software Testing adequate?
3	In your view, is the subject sufficient to understand Software Testing?
4	During the course, did teaching approaches keep you motivated to learn?
5	How useful were the expository classes discussed for the learning of the subject?
6	How useful were the activity lists for learning the subjet?
7	How useful were Dojo Randori's activities with Scratch for the learning of the subject?
8	How useful were Dojo Kake's activities for learning the subject?
9	How useful were the Case Study activities for learning the subject?
10	Would the subject benefit if it was done face-to-face?

In the first part of the questionnaire entitled "Student's perception of the subject", questions were asked to be answered on the Likert Scale from 1 to 5, where the scale's attributions were: 1 - Strongly disagree, 2 - Disagree, 3 - Neutral, 4 - Agree and 5 - Strongly agree. Aspects of the subject itself were initially evaluated: relevance, adequacy, attractive, motivation, the teaching approaches used, with the exception of discussion of practical projects, as they were carried out by invited professors. The results can be seen in Figure 2 and then there is a discussion about the results.

The content taught about Software Testing was considered relevant to be learned, with 100% positive responses ("Strongly agree" or "Agree" answers) by all students of the subject. The students considered the topics and tools used of extreme quality and that there was a balance between the teaching of theory and practice so that the performance of the activities could be carried out in a practical way and with a strong theoretical basis.

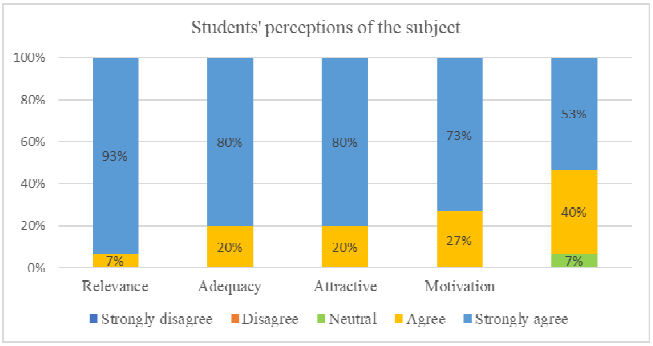

Fig. 2. Results of students' perceptions of the subject.

The teaching approaches in conjunction with the content taught on Software Testing were considered adequate by the students, with 100% positive responses by all students of the subject. The students considered that the approaches reinforced learning when working in a team, where there was a need for each one to handle what was being learned. The students commented that the Dojo activity increased the engagement in the practice itself, as it evoked the desire to fulfill mission as quickly as possible. They also commented that the approach was thought-provoking, exciting and fun.

The subject was considered sufficient by the students to learn Software Testing, with 100% positive responses by all students of the subject. The students commented positively on the exchange of information and experiences among the students, commented on the questionnaire that the classes were attractive and even the slides (materials) were not very full of content. The examples and tips were essential for learning, and the participation of the students for the exchange knowledge was essential.

Throughout the subjects, 93% of students considered that teaching approaches left them motivated to learn. The students considered that although the teaching approaches were mostly satisfactory, they lacked some other form of encouragement. They were cited that there could have been some challenge coming from the job market using PBL (Problem Basead Learning), using some game or even some gamification to stimulate either the competition or collaboration between them.

Still in the questionnaire, questions were also asked to be answered on a Likert Scale from 1 to 5, where the scale's assignments were: 1 - Completely useless, 2 - Not very useful, 3 - Moderately useful, 4 - Very useful and 5 - Essential. The results for each of the approaches used, with the exception of the discussion of a practical project as mentioned above, can be seen in Figure 3 and then there are discussions about the results.

Regarding the approach of dialogued expositive classes by the students, 93% of the students considered it positive. The students considered the format of the class with several professors and the chances of dialogue to be quite enriching, but the students also commented that even if there was a facilitator, it was very difficult to participate actively in the discussion, so they considered remote teaching as an obstacle to the flow of the approach.

The list of exercises approach had the lowest value of positive considerations among the approach, reaching 73% positive mentions. The students considered the practice to be the most affected by remote teaching, as students answered them after one class for comments about it to be made in another class. The students suggested as possible improvements: lists of non-evaluative exercises, accumulating some lists and having a class day of feedback from all and lists of exercise as part of a gamification.

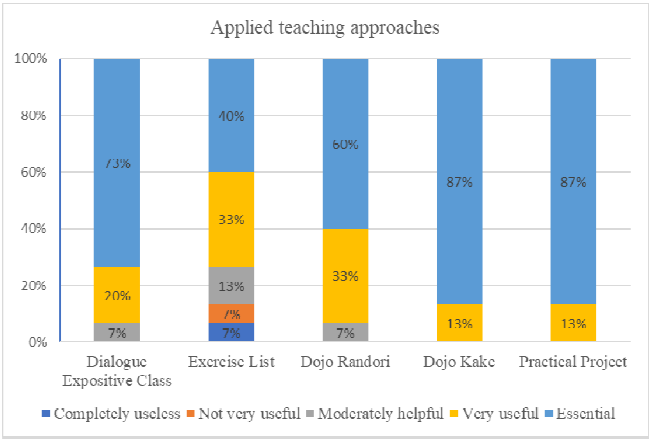

Fig. 3. Results of students' perceptions of approaches.

Dojo randori had a 93% rate of positive considerations. This was the approach that had the most discrepant comments among students. The students commented positively on the collaborative dynamics for learning, the possibility of putting into practice what was learned in theory and the challenge of solving a problem trying to condense everyone's thinking. However, the students considered that not all were able to participate effectively, the initial difficulty of communication and that there was a high gap in the technical knowledge and the tool among the students.

The dojo kake had a 100% rate of positive considerations, with 87% being “Essential” responses. The students highly praised the approach and its operation, even in remote teaching. The students considered that the approach delivered a higher collaborative learning gain than the dojo randori, commented that small group learning was more effective and positively evaluated the anticipated delivery of the challenges the day before the activity for evaluation. Regarding negative aspects, students would like to have had more activities from this approach, use different programming languages and would like to insert a class that could be the real-time evaluation of the task of each team.

The practical project approach, like the dojo kake, had a 100% rate of positive considerations, with 87% of “Essential” responses. The students commented as positive aspects about the autonomy of the teams, the possibility of using the technical knowledge and tools that were obtained during the rest of the subject, in addition to experiencing different moments contained in a Software Testing process. As a negative point, students would like a feedback class on their results and would like there to be more practices of this approach in the subject.

Finally, a question was asked about whether there would be benefits if the subject were carried out in person, to be answered on a Likert Scale from 1 to 5, where the scale's attributions were: 1 - Strongly disagree, 2 - Disagree, 3 - Neutral , 4 - Agree and 5 - Strongly agree. The result can be seen in Figure 4 and then a discussion is made.

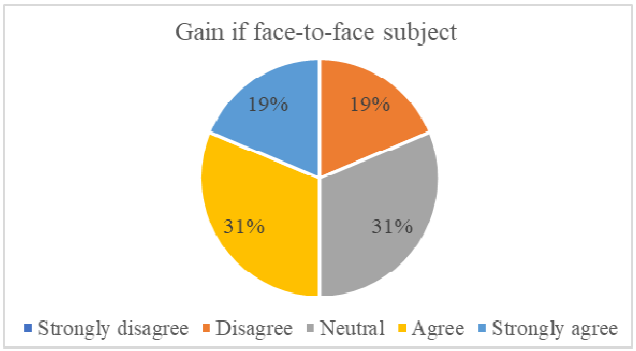

Fig. 4. Perception of earnings if the subject was face-to-face.

Around 50% of the positive responses evaluated that the subject would benefit from being done in person. The students commented that they considered from the form that it was built, there was no damage in the subject of software testing because it was given remotely. However, they stated that the practices of the dialogued expositive class and the dojo randori would have benefited from the face-to-face format to facilitate interaction between all. Finally, the students praised the decision of the classes to have been mostly synchronous, as some students reported difficulties with learning in other subjects, in which, most of the classes were given asynchronously.

VII. Conclusion

The course was designed to be taught face-to-face. However, as soon as there was a need to teach subjects in the form of remote teaching, it was necessary to update several approaches, mainly those considered as active teaching methodologies. A first general adaptation in the entire teaching plan [3] and a continuous flow of feedbacks and adaptations implemented by the authors made this subject manageable. At the end, through a questionnaire, students' ideas were also collected and the level of satisfaction with the subject was verified.

Regarding the RQ surveyed, the response obtained from the students' feedback was positive, it was chosen in addition to knowing whether there was the possibility of adapting a subject to teach Software Testing using active teaching methodologies, also checking the degree of satisfaction with the teaching approaches used. From the results in Section VI, it is seen that the students were satisfied with the learning, however the possibility of gains if the subject was done face-to-face can not be ruled out.

The experience of teaching an adapted subject due to the isolation scenario from the COVID-19 pandemic helped us to identify many good practices and challenges. We hope that this can help to ensure that the learning of students in Computer Science courses, mainly in Software Engineering subjects, is not impaired, due to changes in future pandemics or problems that need social distance.

A. Contribuitions

The main contribution of this work is a set of findings collected from the experience carried out in a subject to teach software testing, which can be used by other computer professor, principalmente em disciplinas de engenharia de software to adapt their teaching plans. A second contribution is the demonstration of the possibility of using more practical teaching approaches supported by active teaching methodologies in subjects that need to be taught in remote education.

B. Limitations

There are some limitations in this work, the main one being a threat of internal validity with regard to the results obtained, since different groups of experiment and control were not used to compare the results. This was due to the priority being the distance due to COVID-19 pandemic. Another limitation is that it is not possible to guarantee that the subjects that use the findings and the teaching approaches described will be successful. The work is about presenting findings that the authors collected, discussed and implemented and became a successful case. Thus, the findings discussed in this work can be used to assist in the decision-making of professors to use a particular teaching approach.

C. Future Works

As future works, the authors intend to replicate the methodology with the teaching approaches in an undergraduate course in computer science. However, the improvements that were collected during the experiment will already be applied to the teaching plan previously. Another possibility is to apply the methodology to different teaching topics in Software Engineering subjects.

Acknowledgements

The authors would like to thank CAPES (Coordination and Improvement of Higher Level or Education Personnel) for granting an institutional Doctoral scholarship to the author of this work. This work belongs to SPIDER Project (http://spider.ufpa.br).

References

[1] L. Marques, P. Matsubara, J. C. de Souza Filho, G. Gomes, B. Gadelha, and T. Conte, “Challenges and Learning from Remote Teaching of Usability and UX: An Experience Report”. 19th Brazilian Symposium on Software Quality, Brazil, 2020.

[2] I. S. Elgrably and S. R. B. Oliveira, "Construction of a syllabus adhering to the teaching of software testing using agile practices". IEEE Frontiers in Education Conference, Sweden, 2020.

[3] I. S. Elgrably and S. R. B. Oliveira, "Model for teaching and training software testing in an agile context". IEEE Frontiers in Education Conference, Sweden, 2020.

[4] S. Ouhbi and N. Pombo, "Software Engineering Education: Challenges and Perspectives". 2020 IEEE Global Engineering Education Conference, Portugal, 2020.

[5] M. K. Keleş and S. A. Özel, “A Review of Distance Learning and Learning Management Systems”. In Virtual Learning. IntechOpen, Rijeka, Chapter 1, 2016.

[6] F. Trinta, P. A. L. Rego, and W. Viana, “Teaching Development of Distributed Software during COVID-19”. presented at the SBES ’20: 34th Brazilian Symposium on Software Engineering, Oct. 2020. doi: 10.1145/3422392.3422480.

[7] V. Lelli, R. M. C. Andrade, L. M. Freitas, R A. S. Silva, F. G. S. Filho, R. F. Gomes, and J. S. O. Severo, “Gamification in Remote Teaching of SE Courses”. presented at the SBES ’20: 34th Brazilian Symposium on Software Engineering, Oct. 2020. doi: 10.1145/3422392.3422497.

[8] T. Kanij and J. Grundy, "Adapting Teaching of a Software Engineering Service Course Due to COVID-19". 32nd Conference on Software Engineering Education and Training, Germany, 2020.

[9] V. Garousi, A. Rainer, P. Lauvås jr, and A. Arcuri, “Software-testing education: A systematic literature mapping”. Journal of Systems and Software, vol. 165, p. 110570, 2020.

[10] J. A. O. G. da Cunha, G. A. Marques, W. L. Lemos, U. D. Câmara Jr, and F. J. S. Vasconcellos, “Software engineering education in Brazil”. XXXII Brazilian Symposium, Brazil, 2018.

[11] L. P. Scatalon, R. E. Garcia, and E. F. Barbosa, "Teaching Practices of Software Testing in Programming Education". IEEE Frontiers in Education Conference, Sweden, 2020.

[12] V. Garousi, M. Felderer, M. Kuhrmann, K. Herkiloğlu, and S. Eldh, “Exploring the industry’s challenges in software testing: An empirical study”. J Softw Evol Proc, vol. 32, no. 8, 2020.

[13] J. Morán, “Changing education with active methodologies”. In: Convergências Midiáticas, Educação e Cidadania: aproximações jovens. Coleção Mídias Contemporâneas, v. 2, n. 1, 2015.

[14] A. Diesel, A. L. Santos Baldez, and S. Neumann Martins, “The principles of active teaching methodologies: a theoretical approach”. Revista Thema, vol. 14, nº 1, 2017.

[15] F.A. Zorzo, D. Nunes, E. Matos, I. Steinmacher, J. Leite, R. M. Araujo, R. Correia, and S. Martins, “Training References for Undergraduate Computing Courses”. Sociedade Brasileira de Computação (SBC), 2017.

[16] MEC, “National Curriculum Guidelines for Undergraduate Computing Courses”. (DCN16), Brazil, 2016.

[17] G, Scallon, “Evaluation of Learning in a Competency Approach”. Curitiba: Pucpress. 2015.

[18] S. Aguilar, C. Holman, and J. Fishman, “Game-Inspired Design: Empirical Evidence in Support of Gameful Learning Environments”. Games and Culture, 2018.

[19] P. Lauvås Jr and A. Arcuri, "Recent Trends in Software Testing Education: A Systematic Literature Review". in The Norwegian Conference on Didactics in IT education, 2018.

[20] B. V. M. Sabin, H.Alrumaih, J. Impagliazzo, B Lunt, M Zhang, B Byers, W Newhouse, B Paterson, S Peltsverger, and C Tang, “Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology”. A Report in the Computing Curricula Series Task Group on Information Technology Curricula. ACM, 2017.

[21] J. Sacristán, A. Gómez, J. Rodríguez, J. Santomé, F. Rasco, and J. Méndez, “Educating by Skills: What's New?”. Artmed Editora. 2016.

[22] A. Ferraz and R. Belhot, “Bloom's taxonomy: theoretical review and presentation of the instrument's adjustments to define instructional objectives”. Gest. Prod., São Carlos, v. 17, n. 2, 2010.

[23] K. Beck, “TDD - Test Driven Development”. 1. Ed. Porto Alegre, Bookman Editora, 2010.

[24] L. Cronbach, “Coefficient alpha and the internal structure of tests”. Psychometrika, n. 16, 1951.